Transformer For Medical AI

Project 6

Zheng Hexing 2023311430
Chang Hwan Kim 2024321234
Maftuna Ziyamova 2024311551
Lee Woo Bin 2025311560

Problem Formulation - Motivation

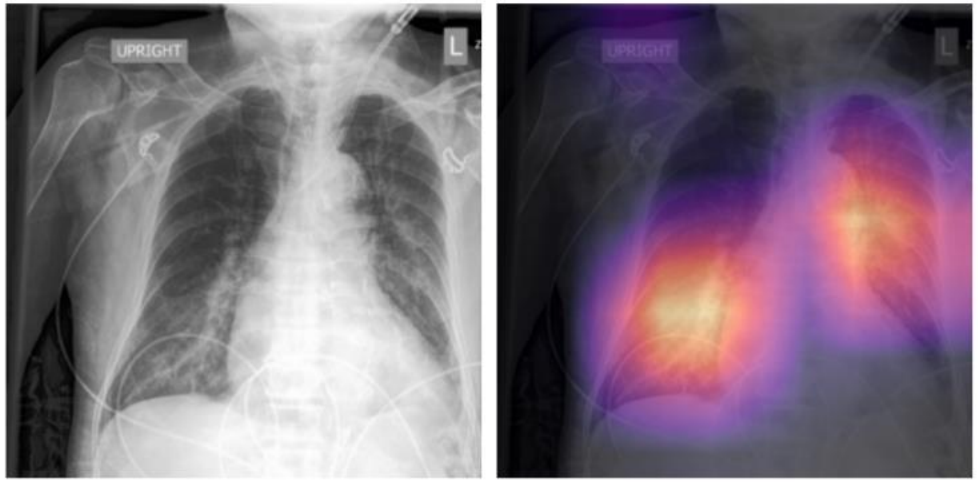
- Chest radiography is the most frequently performed imaging examination globally
- Essential for screening, diagnosing, and managing numerous life-threatening conditions
- Significant potential for automated interpretation systems to match or exceed radiologist accuracy

YONSEI UNIVERSITY

Problem Formulation - Goal

- Develop a Transformer-based model capable of accurately diagnosing chest radiographs based on 14 labeled observations

- Generate interpretable heatmap visualizations highlighting model attention areas to support clinical decision-making

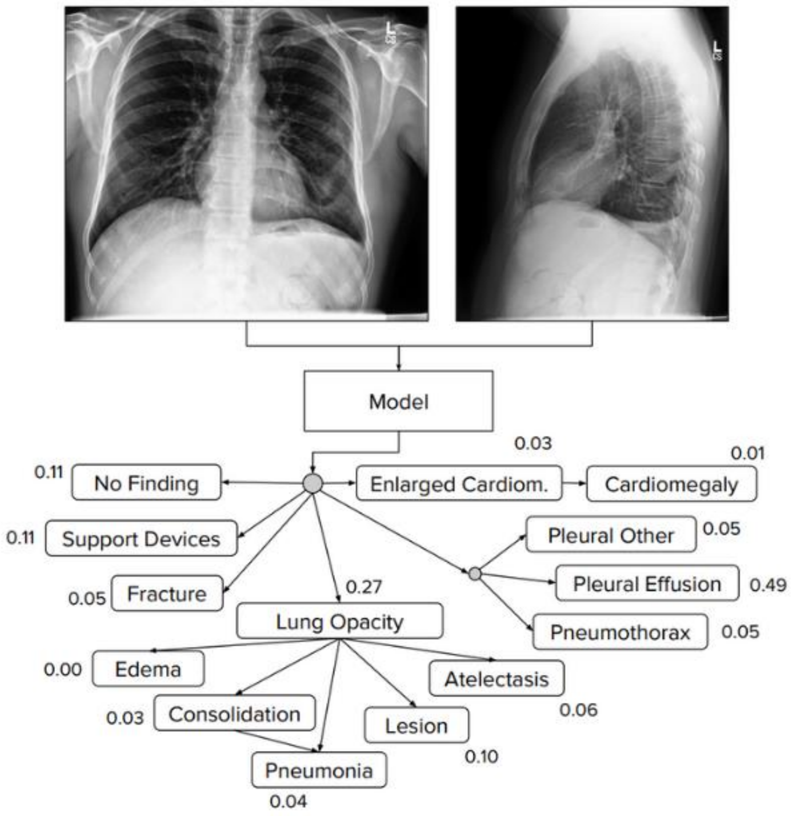


YONSEI UNIVERSITY

Data and Methods

Dataset: CheXpert

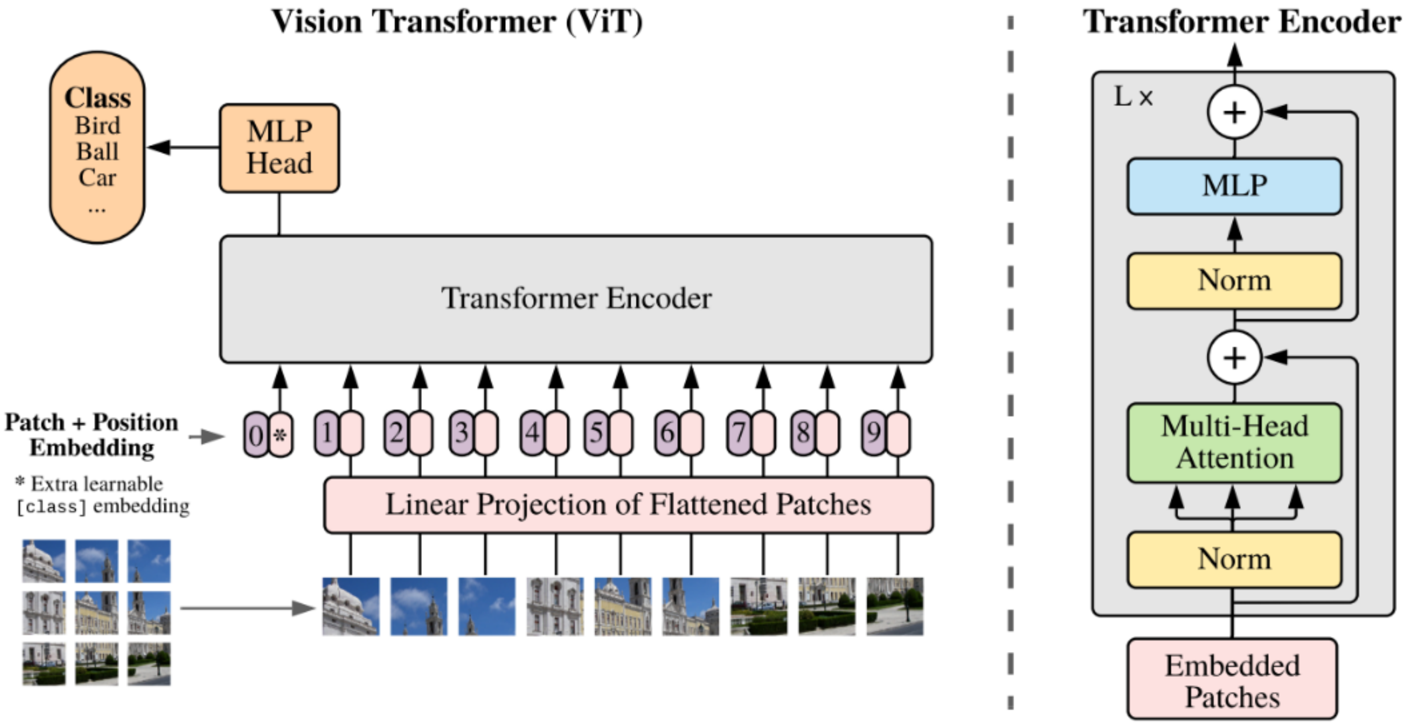
- 224,316 chest radiographs from 65,240 patients
- 14 labeled clinical observations
- Automated labeling system designed to detect and classify observations, including inherent uncertainties
- Validation set of 200 radiographic studies that was manually annotated by 3 board-certified radiologists
- *We mostly used **<u>CheXpert-v1.0-small</u>** that is a smaller, downsampled version of the original dataset and it would be explained why later*



YONSEI UNIVERSITY

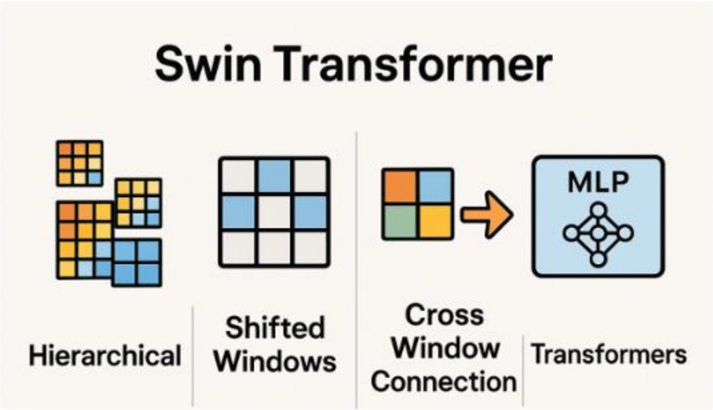
Data and Methods

We used 3 different architectures: Vision Transformer (ViT), Swin Transformer, BEiT Transformer

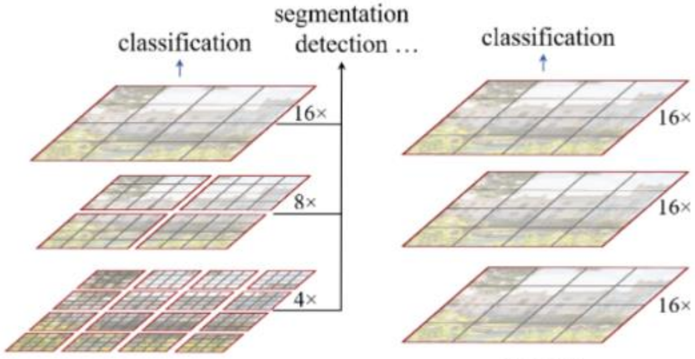


Vision Transformer (ViT) Overview

Data and Methods

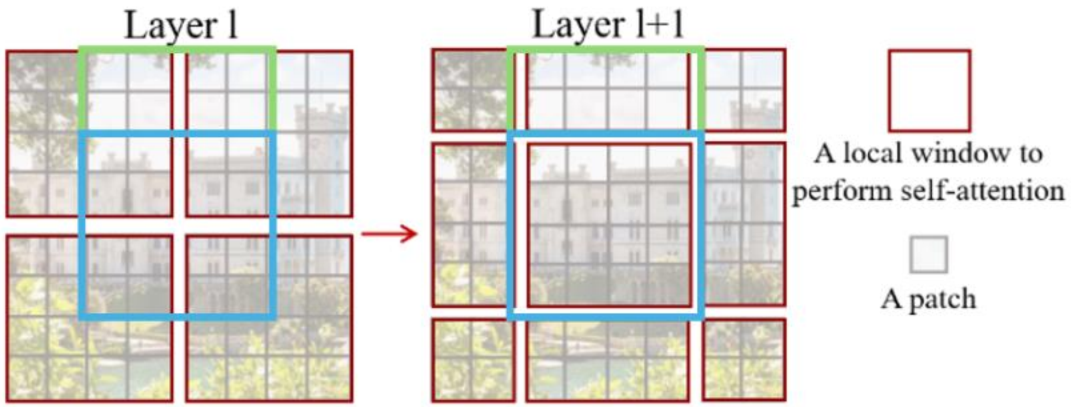


Hierarchical architecture:



(a) Swin Transformer (ours) (b) ViT

Cross window connection and Shifted Windows :

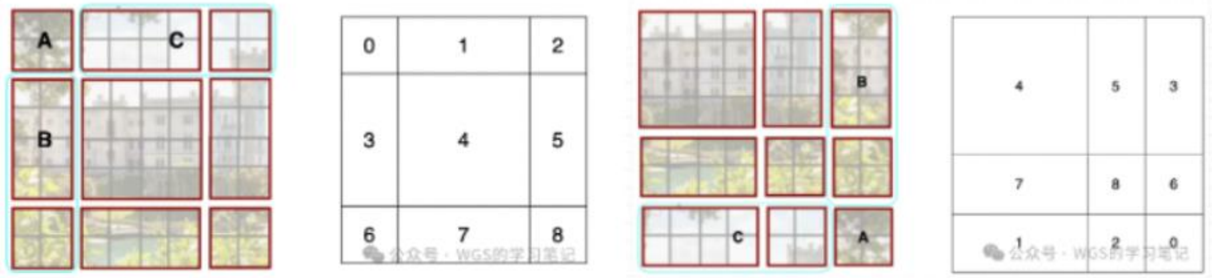


Shifted Windows

- Improved global modeling
- Better context aggregation
- Preserves locality

Cross window connections

- Stronger contextual learning
- Better spatial coherence
- Handles object boundaries well

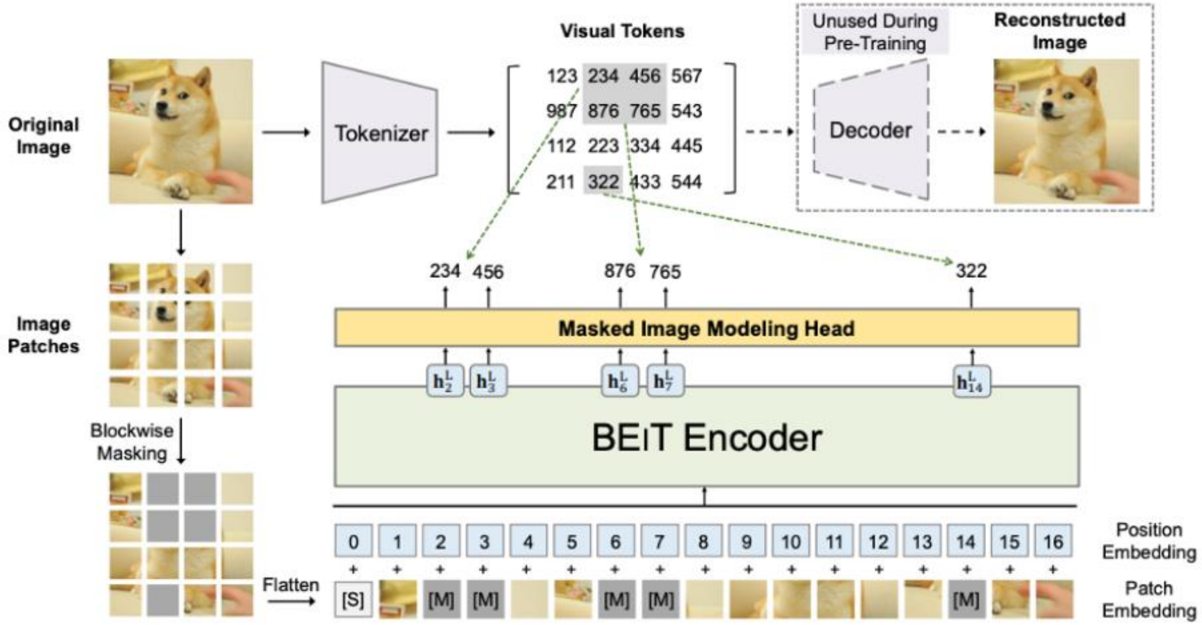


YONSEI UNIVERSITY

Data and Methods

- **BEiT Transformer** model is a vision transformer based on self-supervised learning
- It applied BERT's Masked Language Modeling to images by predicting visual tokens

- It uses a Discrete VAE to convert images into codebook-based visual tokens and predicts the masked parts
- BEiT shows strong pretraining performance and is effective for various downstream vision tasks

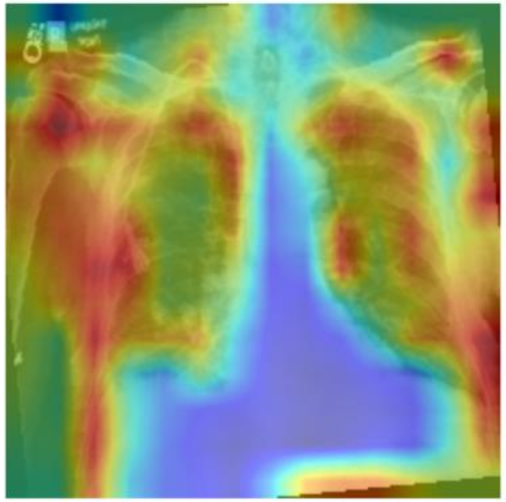


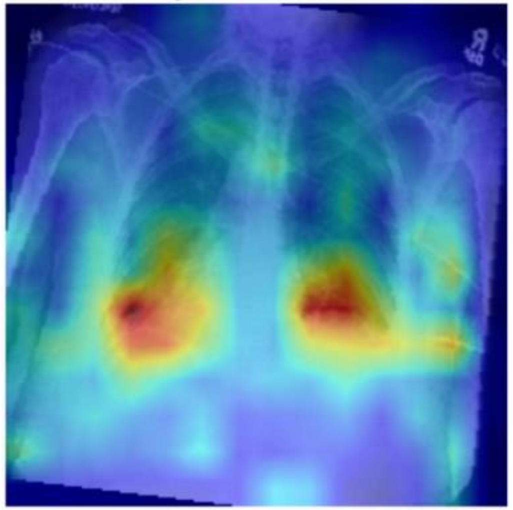
YONSEI UNIVERSITY

Data and Methods

- Last layer of the model's CLS token – trained to gather "information that represents the entire image" during the learning process
- Converting "the attention score value that the CLS token gives to each patch in the image" into "a 2D heatmap"
- Using a **heatmap**, we can understand intuitively where the Vision Transformer model is focusing on the image

Image 1 - ViT Attention





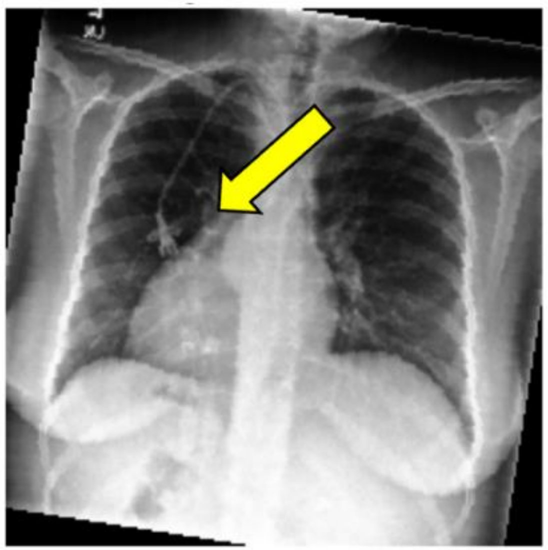
An example of 2D heatmap from our previous baseline model
: since the model is **not fully trained** yet, it's **not accurate**

An example of 2D heatmap our current baseline model
: the model **properly focuses** on the part used to determine the disease

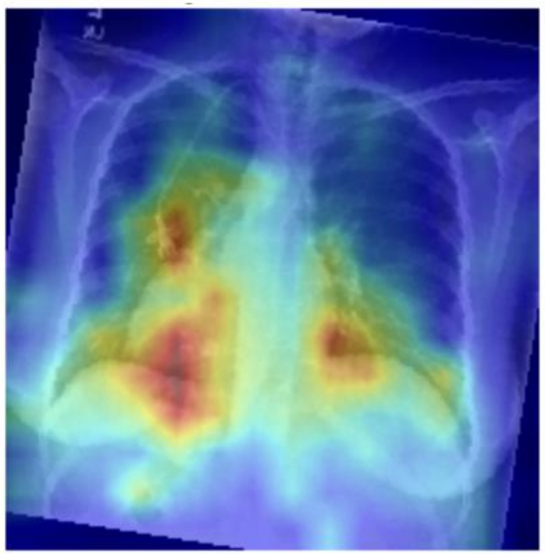
YONSEI UNIVERSITY

Data and Methods

- Left Image: Original chest radiograph. The yellow arrow indicates the presence of a support device.
- Right Image: Heatmap highlighting the model's area of attention, precisely aligning with the support device's location.



Original image

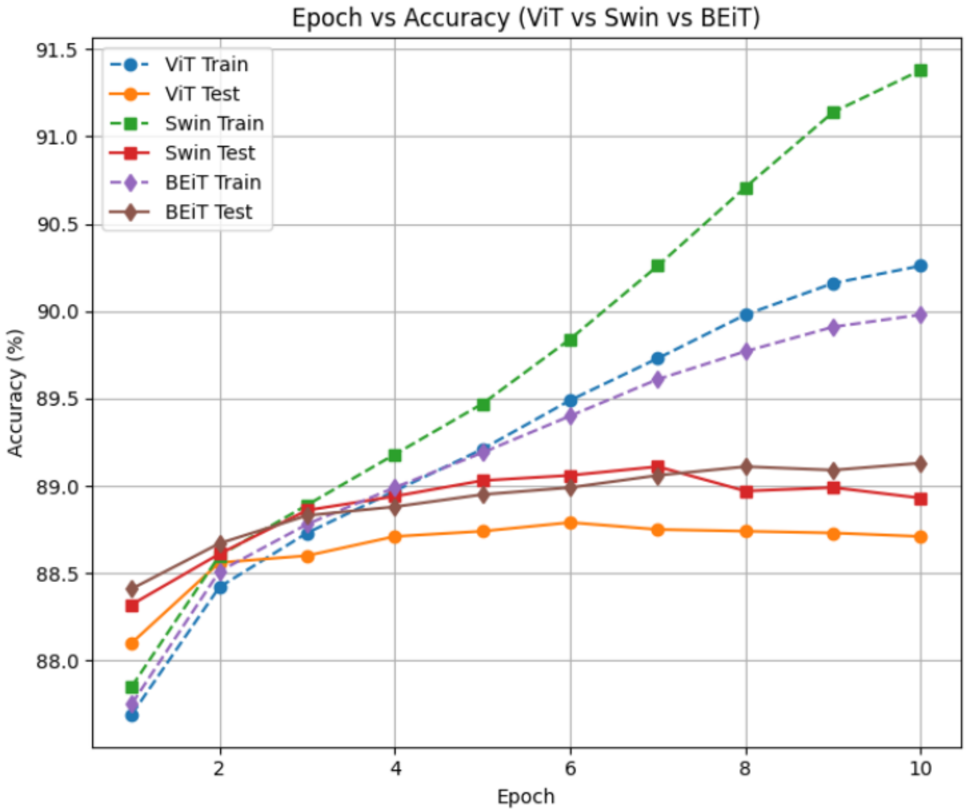


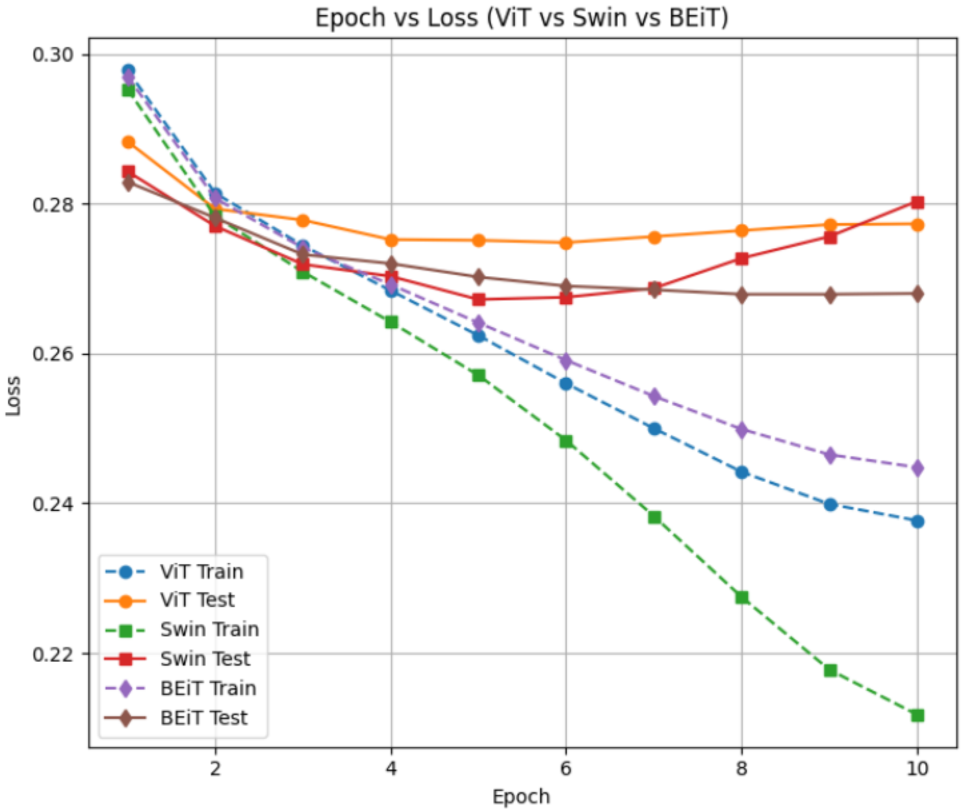
Heatmap image

Another example of 2D heatmap from our baseline model

: correctly classifies support device

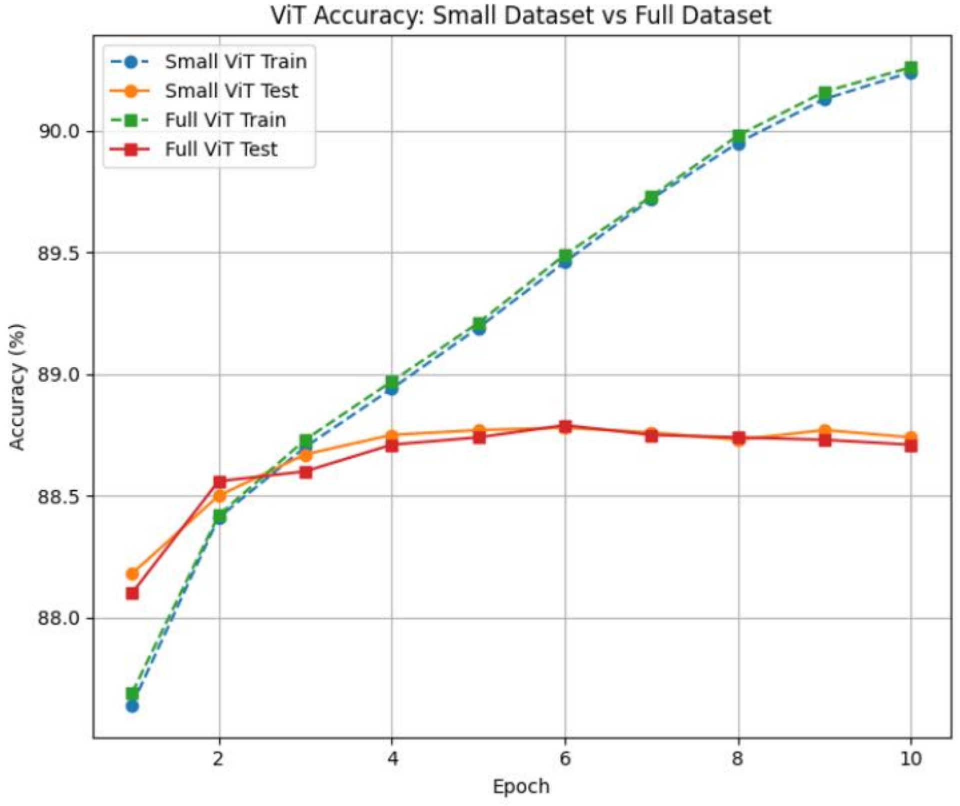
Results

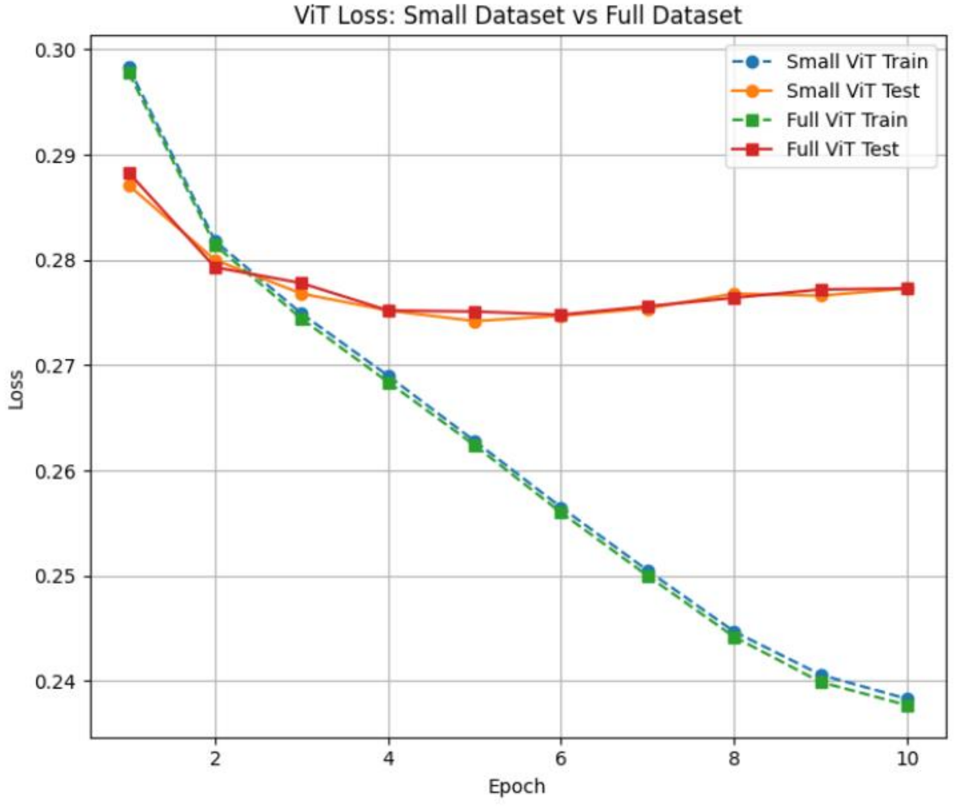




- BEiT transformer performs best among those models

Results





- No difference in results between training full and light datasets

Discussion and Future Work

- Evaluated transformer architectures (ViT, Swin, BEiT) on CheXpert dataset.
- Compared performance using both full and reduced datasets.
- **Why BEiT Performs Best:**
  - BEiT leverages masked-image modeling for pre-training, enhancing its ability to generalize and handle diverse image features.

- Visualization and comparison of attention maps across different transformer models will be included on our GitHub.
- Conduct a simple robustness test by adding small Gaussian noise or perturbations to the input images, evaluate performance degradation, and perform additional training to increase robustness if needed.

Thank you for attention!